AD-A039 795 COLLEGE OF WILLIAM AND MARY WILLIAMSBURG VA F/G 9/2
TIMING FORMULAS FOR DISSECTION ALGORITHMS ON VECTOR COMPUTERS.(U)
MAY 77 W G POOLE N00014-75-C-0879
UNCLASSIFIED TR-14 NL

1 OF 1
AD A039795

END
DATE FILMED
6-77

(12) NW

# Timing Formulas for Dissection Algorithms on Vector Computers

W. G. Poole, Jr.

Technical Report 14

May 1977

# TIMING FORMULAS FOR DISSECTION ALGORITHMS ON VECTOR COMPUTERS

W. G. Poole, Jr.
College of William and Mary

## SUMMARY

The use of the finite element and finite difference methods often leads to the problem of solving large, sparse, positive definite systems of linear equations. Recently the one-way dissection and nested dissection algorithms have been developed for solving such systems. Concurrently, vector computers (computers with hardware instructions that accept vectors as operands) have been developed for large scientific applications. In reference 1, George, Poole and Voigt analyzed the use of dissection algorithms on vector computers. In that paper, MACSYMA played a major role in the generation of formulas representing the time required for execution of the dissection algorithms. In the present paper the author describes the use of MACSYMA in the generation of those formulas.

## DISSECTION ALGORITHMS

When finite difference or finite element methods are used for approximating solutions of partial differential equations, it is often the case that a large, sparse, positive definite system of linear equations,

$$Ax = b \qquad (1)$$

must be solved. We shall assume that the domain over which the differential equation is defined is a square region covered by an $n$ by $n$ grid consisting of $(n-1)^2$ small squares called elements. It follows that $A$ is an $n^2$ by $n^2$ matrix. The ordering of the unknowns at the grid points determines the location of the nonzero components of $A$ and, consequently, the storage and time required to solve the linear system by Gauss elimination.

An ordering of the unknowns called one-way dissection is due to George (see ref. 2). Referring to figure 1, the idea of one-way dissection is first to divide the grid with $m$ horizontal separators. The unknowns in the $m+1$ remaining rectangles are numbered vertically toward a separator and then the

---

This paper was prepared as a result of work supported in part under NASA Contract No. NAS1-14101 at ICASE, NASA Langley Research Center, Hampton, VA 23665 and in part by Office of Naval Research Contract N00014-75-C-0879.

separator nodes are numbered. The problem is to derive formulas for storage and timing requirements and to minimize those formulas with respect to m (see ref. 2).

The second dissection scheme is called nested dissection (again, see ref. 2) and has been shown to be asymptotically optimal (see ref. 3). The idea here is to divide the grid with both horizontal and vertical separators as shown in figure 2. Unknowns in regions 1 - 4 are numbered before those on separators 5 - 7. Each of the regions 1 - 4 is a square and may itself be dissected using horizontal and vertical separators. Thus the idea may be applied recursively and, in the case $n = 2^k - 1$, nested dissection will terminate after k-1 steps.

Although both dissection orderings were analyzed in reference 1, only nested dissection will be discussed further here because it is a more important algorithm and the generation of its timing formula was a much more formidable task.

The nested dissection algorithm is nontrivial to describe in detail. It was first developed and analyzed with scalar computers in mind by A. George in the early 1970's. The first attempts at obtaining a timing formula were done by hand and only gave a description of the asymptotic behavior, $0(n^3)$. Later, the first few terms were generated by hand. Then in reference 3, A. George obtained the entire formula with the aid of ALTRAN.

## VECTOR COMPUTERS

The existence of vector computers, i.e., computers with hardware instructions that operate on vectors rather than scalars, raises the question of how effective the dissection techniques are on this rather new class of computers. It is assumed that these computers have basic vector instruction execution times which are of the form

$$T_*(j) = S_* + jP_* \ , \qquad (2)$$

where $T_*(j)$ is the total time for the vector instruction *; $S_*$ is an overhead time, called "start-up" time; $P_*$ is the "per-result" time of that instruction; and $j$ is the length of the vector.

The large value of $S_*/P_*$ on currently available vector computers implies that one pays a significant penalty for operation on short vectors; consequently, one would prefer algorithms which permit the longest possible vectors (see ref. 4). However, both of the dissection algorithms work by repeated subdivision of the grid until a minimum operation count is obtained. It is this apparent conflict between the cost of using shorter vectors and the corresponding lower operation counts that was studied in reference 1.

## GENERATION OF FORMULAS

In reference 1, George, Poole and Voigt were interested in obtaining parameterized versions of the timing formulas for the dissection algorithms on vector computers. Such formulas were needed in order to study the effects of varying several parameters. They identified nine parameters characterizing the vector computers: 3 start-up times for vector addition, multiplication, and inner product; 3 per-result times for the same instructions; and 3 scalar operations. Furthermore, there was a parameter, n, related to the problem size and another, $\ell$, related to the algorithm which the user could vary at liberty. The goal was to choose $\ell$ so as to minimize the timing formula for a given set of computer parameters and a given problem size. Obtaining the timing formulas was useful in several ways:

(1) With the formulas in hand, one could study the effects of changing values for the parameters. In a hypothetical sense one could try to optimize subject to certain side constraints. In a very practical sense, manufacturers announced changes in the parameter values several times;

(2) There are several options in the implementation of the dissection algorithms. For example, one can use a vector inner product or a vector "outer product" version (see ref. 1). The choice reduces to comparing the time required for a vector inner product versus a vector addition plus a vector maltiplication. Timing formulas permitted analysis of such options;

(3) Considerable insight into the vectorization of algorithms was gained. For example, average vector lengths could be studied;

(4) Without the formula, a table of timing values for particular choices of the parameters could be generated by executing a model of the algorithm. However, the coefficients in the formulas could not be generated.

The nested dissection timing formula was generated in the following manner. The execution of the nested dissection algorithm was simulated in a top-down fashion. The top level, level 1, involved several summations of which

$$\sum_{i=1}^{j-1} (2^i - 2)^2 \theta\left(\frac{n - 2^i + 1}{2^i}, \frac{4(n+1)}{2^i}, 4\right) \qquad (3)$$

is typical, where $\theta$ is a procedure at the second level. Each of the second level procedures called several third level procedures, e.g.,

$$\text{THETA}(Q,P,K) := \text{CHLSKY}(Q) + P\ \text{LOWSOL}(Q) + \text{MODNES}(Q,P,K) \quad . \qquad (4)$$

CHLSKY, LOWSOL and MODNES are three of the third level procedures defined to be the timing formulas for simple numerical computations, e.g.,

$$\text{CHLSKY}(Q) := \frac{(PA + PM)\,Q^3}{6} + \frac{(SA + SM + PM)\,Q^2}{2}$$

$$+ \left(DSR + \frac{SM}{2} - \frac{SA}{2} - \frac{2\,PM}{3} - \frac{PA}{6}\right) Q - SM \qquad (5)$$

is the timing formula for the factorization of a dense linear system. These third level procedures were formulas for factorization, lower solve and upper solve of dense systems and banded systems and matrix modifications of the form

$$A := A - UVW^T \quad . \qquad (6)$$

Finally, the bottom level consisted of the parameters which characterize the vector computer. E.g.,

$$SA + Q\,PA \qquad (7)$$

is the time for a vector add of length Q.

The second and third levels each consisted of 10 to 15 modules and level 4 consisted of 9 instruction parameters, 1 parameter related to the algorithm and 1 related to the grid size for the problem. The top level module contained several MACSYMA sums of the form

```
SUM(''(EV(((2^I-2)^2)*(THETA((N-2^I+1)/(2^I),4*(N+1)/(2^I),4)),
                                                              (8)
                    EXPAND)),I,1,J-1)  .
```

This is the MACSYMA form of the sum in eq. (3). The entire generated formula consists of over 200 terms and can be found in Appendix B of reference 1. The formula was checked by evaluating it for several sets of parameter values and comparing the results to execution times of a FORTRAN simulation of the algorithm. The one-way dissection formula was generated in a similar, but much more forward, manner.

## CONCLUDING REMARKS

MACSYMA has been shown to be of considerable value in the study of the performance of the nested dissection algorithm when used on hypothetical vector computers. The derived timing formulas lead to an understanding of the effects of varying the parameters which characterize the computers. Options in the algorithm's implementation can be studied as well as the extent to which the algorithm vectorizes.

## REFERENCES


1. George, A.; Poole, W. G., Jr.; and Voigt, R. G.: Analysis of Dissection Algorithms for Vector Computers. ICASE Report No. 76-17, June 8, 1976. (Available as NASA CR-145177.)

2. George, A.: Numerical Experiments Using Dissection Methods to Solve n by n Grid Problems. SIAM J. Numerical Analysis, vol. 14, no. 2, Apr. 1977, pp. 161-179.

3. George, A.: Nested Dissection of a Regular Finite Element Mesh. SIAM J. Numerical Analysis, vol. 10, no. 2, Apr. 1973, pp. 345-363.

4. Lambiotte, J. J., Jr.; and Voigt, R. G.: The Solution of Tridiagonal Linear Systems on the DCD STAR-100 Computer. ACM Trans. Math. Software, vol. 1, no. 4, Dec. 1975, pp. 308-329.

FIGURE 1. - ONE-WAY DISSECTION WITH ORDERING OF UNKNOWNS INDICATED BY NUMBERS (m = 3).

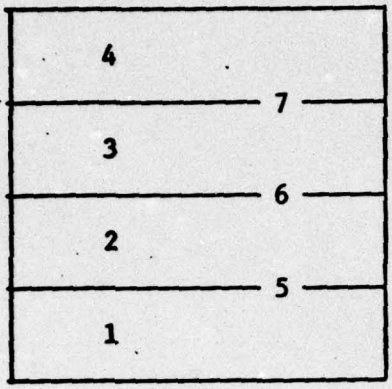
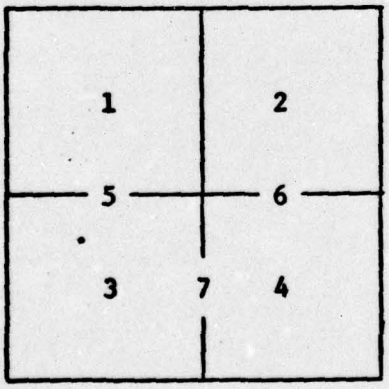



FIGURE 2. - ONE STEP OF NESTED DISSECTION WITH ORDERING OF UNKNOWNS INDICATED BY NUMBERS.



Unclassified

Security Classification

## DOCUMENT CONTROL DATA - R & D

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*


1. ORIGINATING ACTIVITY (Corporate author)

College of William and Mary
Department of Mathematics and Computer Science
Williamsburg, Virginia 23185


2a. REPORT SECURITY CLASSIFICATION: Unclassified

2b. GROUP

3. REPORT TITLE

(6) Timing Formulas for Dissection Algorithms on Vector Computers

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report 14, May 1977


5. AUTHOR(S) (First name, middle initial, last name)

(10) W. G. Poole, Jr


(12) 9 p.

(14) TR-14

| 6. REPORT DATE | 7a. TOTAL NO. OF PAGES | 7b. NO. OF REFS |
|---|---|---|
| (11) May 1977 | 6 | 4 |

8a. CONTRACT OR GRANT NO.

(15) N00014-75-C-0879, NAS1-141 01

b. PROJECT NO.

NR044-459

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

(9) Technical Report 14

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release;
distribution unlimited

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Mathematics Program
Office of Naval Research
Arlington, Virginia 22217

13. ABSTRACT


The use of the finite element and finite difference methods often leads to the problem of solving large, sparse, positive definite systems of linear equations. Recently the one-way dissection and nested dissection algorithms have been developed for solving such systems. Concurrently, vector computers (computers with hardware instructions that accept vectors as operands) have been developed for large scientific applications. In reference 1, George, Poole and Voigt analyzed the use of dissection algorithms on vector computers. In that paper, MACSYMA played a major role in the generation of formulas representing the time required for execution of the dissection algorithms. In the present paper the author describes the use of MACSYMA in the generation of those formulas.


DD FORM 1 NOV 68 1473 (PAGE 1)

S/N 0101-807-6801

Unclassified

Security Classification

379 000

| 14. KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| NESTED DISSECTION | | | | | | |
| ONE-WAY DISSECTION | | | | | | |
| FINITE ELEMENT SYSTEMS | | | | | | |
| VECTOR COMPUTERS | | | | | | |
| MACSYMA | | | | | | |